

# Milestone One

## Automotive Grade Linux Documentation Assessment

*by Scotty's Documentation Services, INC*

## Executive Summary

Scotty's Documentation Services has conducted an investigation of Automotive Grade Linux (AGL) user documentation found at <http://docs.automotivelinux.org/> and <https://wiki.automotivelinux.org/> and has found several areas for improvement. We examined the functionality of the website, the organization and existence of content, and the overall quality of documentation. We found deficiencies in website navigation and organization, documentation architecture, content coverage, and writing styles and quality. This paper discusses each of these areas, provides some examples, and proposes some solutions.

## Problems & Solutions

- **Doc Web Page Design:** The landing page for documentation (<http://docs.automotivelinux.org/>) is redundant, has poor navigation, and is not architected well. Aside from the landing page, subsequent pages suffer from the same problems -- poor navigation and redundancy.

**EXAMPLE:** The Navigational buttons along the top of the documentation landing page do not provide good context. For example, the "Architecture Guides" button would seem to indicate the area where the reader can find separate guides on architectures. Organized under "Architecture Guides" is a large specification and an extensive security blueprint. These documents are not "Guides".

Another example is the "Development Guides" page contains non-guides (e.g. Yocto) as well as a sub-manual whose name is "Guides".

The landing page for the AGL documentation contains wasted real estate in the main window. It also contains redundant "Contribute" buttons. And, the "Search" button is not self-clearing and unable to navigate through returned results.

For sub-pages of the doc site, the navigation pane has topics that cannot be collapsed. This makes for a very long and cluttered list of navigational topics. The primary window of these web pages also have repeated versions of the navigation pane topics scattered throughout. This primary window should be for content, not navigation.

**SOLUTION:** Redesign the doc landing page and the scheme used for sub-pages. See the attached two appendices with discussion for proposed website solutions.

- **Content Not Well Architected:** Information falls into three general categories: conceptual, procedural, and referenced. With large documentation sets, you have options for architecting the information. You can gather like information into areas to create a learning environment that sets the reader up with expectations regarding what kind of information they are going to find in any given area (i.e. reference manuals, “how-tos”, and concepts). You can group all types of information for a given topic under one cover. An example of that can be a single manual on Widgets. Inside that single manual all the information for Widgets exist: introductory/conceptual, “how-to” information for widgets, and any widget reference information.

**EXAMPLES:** As the information stands, it is scattered about in several different areas in such a manner that it is hard to predict what you might see. “Getting Started” has a book called “Getting Started” as well as several other books. One of the “Getting Started” documents is really just a pictorial reference of the AGL home screen. “Troubleshooting” seems more like a “how-to” set of topics. Is the SDK title under “Getting Started” really a getting started item or more of a user manual on a larger topic?

**SOLUTION:** Categorize all the topics/books into a type of information. Decide on an organizational strategy and stick with that for all areas. Base your top-level navigation on areas of information that satisfy the user. That could be “by topic” (e.g. all information under a specific topic). Or, that organization could be by information type (e.g. a left NAV pane whose high levels are organized around types of information). Use appropriate and consistent naming for grouped information, titles, topics, and manuals.

- **Content Gaps:** Gaps exist in the following area:
  - ◆ Consistent use of overview material that prefaces a manual.
  - ◆ Use of directional text to help guide the reader through a large group of manuals or topics. This type of information can also include consistent application for directing a user to outside sources of information.
  - ◆ Glossary information sporadically applied (e.g. acronym definitions).
  - ◆ Migration information for readers moving from one major release to the next.
  - ◆ Consistent use of machine output (or results) given example commands or code.
  - ◆ Conceptual information.
  - ◆ Use cases.

**EXAMPLES:** Some titles use an “Abstract” heading that either summarizes some concepts or tells the user what to expect from a particular document or topic. The “Developer Guides” is a good area to see the inconsistent use or lack of the “Abstract”. (See [http://docs.automotivelinux.org/docs/devguides/en/dev/reference/host-configuration/docs/0\\_Abstract.html](http://docs.automotivelinux.org/docs/devguides/en/dev/reference/host-configuration/docs/0_Abstract.html), [http://docs.automotivelinux.org/docs/devguides/en/dev/reference/sdk-devkit/docs/part-1/1\\_0\\_Abstract.html](http://docs.automotivelinux.org/docs/devguides/en/dev/reference/sdk-devkit/docs/part-1/1_0_Abstract.html), <http://docs.automotivelinux.org/docs/devguides/en/dev/reference/ctrlr/controller.html>.)

See the “Host Configuration” manual under “Developer Guides” for an example of missing introductory/directional text to prepare the reader for the manual. The “Abstract” presents a short purpose statement for the “section”. No real help in understanding further what to expect from this “Host Configuration” manual. (See [http://docs.automotivelinux.org/docs/devguides/en/dev/reference/host-configuration/docs/0\\_Abstract.html](http://docs.automotivelinux.org/docs/devguides/en/dev/reference/host-configuration/docs/0_Abstract.html)).

See the “Part 1 - Hardware” section in the “Acronyms and Abbreviations” section all within the “Architecture Guides” area for some definitions. No other documents in the AGL set other than this area attempts any term definitions. (See [http://docs.automotivelinux.org/docs/architecture/en/dev/reference/security/part-1/0\\_Abstract.html](http://docs.automotivelinux.org/docs/architecture/en/dev/reference/security/part-1/0_Abstract.html)).

We could not find any migration information.

See “Host Configuration” under “Developer Guides” for some examples of shell commands with and without supporting output. (See [http://docs.automotivelinux.org/docs/devguides/en/dev/reference/host-configuration/docs/1\\_Prerequisites.html](http://docs.automotivelinux.org/docs/devguides/en/dev/reference/host-configuration/docs/1_Prerequisites.html)).

See “Yocto Layers” under “Developer Guides” for an example of missing conceptual information. Do all users know what “Yocto” is and its role? (See <http://docs.automotivelinux.org/docs/devguides/en/dev/reference/AGL.html>).

The existing set of written documentation does cover some specific uses. Obviously, there are missing cases as indicated by the “improvements for Documentation” wiki. We can’t assess how many “use-cases” are needed. But knowing that the set is incomplete is a start.

**SOLUTION:** For every manual, create a consistent introductory section that describes what the reader can expect to learn from the manual, perhaps some organizational breakdown of the manual, and perhaps some areas where other related types of information can be found.

Decide how you want to address terms used throughout the documentation. Do you want a glossary for the entire manual set? Do you want a glossary per manual? Do you simply want to define each term as you go?

Does moving from one major AGL release to the next present problems? Is it seamless? If users need to do special things to migrate along the release path, decide how you want to present that information? Maybe you have that information in release notes. Maybe it is a reference manual with a title of “Migration”.

Most examples currently use a special font for shell commands. Some show output, while others do not. It would be good to have a consistent policy on what type of output and how much to show. Readers unfamiliar with some commands like the reassurance of some output to tell them they are doing things correctly. If new output is provided, then a summary of successful results is helpful. Also, providing instruction on what might go wrong can be helpful. The point here is to develop a consistent strategy for this type of “how-to” text.

Conceptual information is important as a base for understanding tasks. Concepts need to be provided for all topics -- some more than others. For example, when explaining architecture, more conceptual information is needed than when explaining how to perform a task. Some projects group all the conceptual information under one area (e.g. Yocto Project documentation). Some present concepts at the front-end of a topic. A strategy needs to be decided on and then consistently followed. But the key is that conceptual information needs to be there for all topics. The exception is pure reference material (i.e. glossaries, method descriptions, and so forth).

Use cases should be addressed as topics in a “Developer Guide” or, in the case of very large tasks a complete manual. Another possibility is to group “Use Cases” into a separate area unto themselves.

- **Poor Writing consistently throughout the documentation:**

Grammar and spelling errors exist throughout the documentation. There seems to be a marked difference between the wiki and the general documents. The writing in the wiki seems to be in better shape.

All writing should follow accepted technical writing standards (i.e. active voice, concise sentence construction, consistent formatting, and so forth).

**EXAMPLE:** *For the components of the application framework, built top of the security framework, instead of pulling the huge set of packages from Tizen, we decided to refit it by developing a tiny set of components that would implement the same behaviour but without all the dependencies and with minor architectural improvements for AGL. (See [http://docs.automotivelinux.org/docs/apis\\_services/en/dev/reference/af-main/0-introduction.html](http://docs.automotivelinux.org/docs/apis_services/en/dev/reference/af-main/0-introduction.html).)*

**SOLUTION:** All writing needs to go through someone or group responsible for final copy. This is likely a methodology change or change of mindset to apply effort to the overall quality of the writing and presentation.

- **Inconsistent Voicing:** It appears documentation is being sourced across many different people. When this happens, you get a “fragmented” feel to the manuals. It also creates a situation where you have varying levels of quality depending on the contributor’s writing skill. The most common issue you find is some authors sticking to the “us” and “we” phrasing, while other authors go with the better “you” phrasing. Some people mix and match. A consistent voice or “feel” should exist across all documentation. Additionally, “active” and “past tense” are mixed between authors as some prefer one over the other.

**EXAMPLE:** *We will cover topics starting from the lowest level (Hardware) up to the highest levels (Connectivity and Application).* (See <http://docs.automotivelinux.org/docs/architecture/en/dev/reference/security/README.html>.) *Once you run `aglsetup.sh` with your desired parameters, you can build any target desired.* (See [http://docs.automotivelinux.org/docs/getting\\_started/en/dev/reference/source-code.html](http://docs.automotivelinux.org/docs/getting_started/en/dev/reference/source-code.html).)

**SOLUTION:** Decide how you want your writing to come across to your target audience and stick with that. Fixing this falls into the same area as fixing poor writing in general. A person or set of people need to be responsible for the quality of the final documentation products.

Part of the solution is an active documentation style guide. This guide is a good source for environments that have multiple authors whose primary job is not writing.

- **Going Forward / Maintenance:** Following are tasks to move forward. Some things depend on prior tasks while some tasks could be simultaneously addressed:
  - ◆ Complete Milestone One (M1) as defined by the Statement of Work (SOW).
  - ◆ Complete M2 as defined by the SOW.
  - ◆ Complete M3 as defined by the SOW.
  - ◆ Redesign and completion of the AGL documentation website to give it improved content organization and navigation.
  - ◆ Decide on and move forward with a general content architecture scheme regarding new and existing documentation. Factor in the role of the wiki.
  - ◆ Document additional use-cases beyond those listed for M2 and M3 of the SOW.
  - ◆ Address documentation methodology regarding authoring and QA activities.
  - ◆ Re-architect the documentation set. This activity could involve consolidation, segregation, and replacement of manuals/topics. Factor in the role of the wiki.
  - ◆ Scrub the text for standard technical writing quality (grammar, spelling, brevity, voicing).

# AGL Documentation Landing Page

Following is a representation framework for a proposed AGL Documentation Landing Page at <http://docs.automotivelinux.org/>.

AUTOMOTIVE  
GRADE LINUX



Automotive Apps with HTML5, Qt OpenGL  
Multiple platforms on a common code base  
Free and open source

EEL 5.1.0



Select your chosen release in the dropdown menu and then select a category to access its documentation.  
You can visit the [AGL Wiki](#) for the latest News & Updates and to Get Involved in the AGL Community.

Getting  
Started

Architecture  
Guides

Hardware  
Support

Developer  
Guides

APIs &  
Services

# AGL Documentation Sub-Pages

Following is a representation framework for a proposed AGL Documentation Sub-Page (i.e. a page navigationally lower than the landing page). For example, clicking on “Getting Started” at

[http://docs.automotivelinux.org/docs/getting\\_started/en/dev/](http://docs.automotivelinux.org/docs/getting_started/en/dev/).

AUTOMOTIVE  
GRADE **LINUX**


- + **Getting Started**
  - + Quick Guides
  - + Homescreen
  - + Troubleshooting
  - + SDK
- + Architecture Guides
- + Hardware Support
- + Developer Guides
- + APIs & Services

## Getting Started Overview

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Lorem ipsum dolor sit amet consectetur adipiscing elit duis. Pharetra et ultrices neque ornare aenean euismod. Scelerisque mauris pellentesque pulvinar pellentesque habitant morbi tristique.

Morbi tristique senectus et netus et malesuada fames ac. Sit amet dictum sit amet justo donec enim diam vulputate. Volutpat commodo sed egestas egestas.

Ornare arcu dui vivamus arcu felis. Sed cras ornare arcu dui vivamus arcu. Imperdiet dui accumsan sit amet nulla facilisi morbi. Arcu dictum varius duis at. Nibh praesent tristique magna sit. Cras tincidunt lobortis feugiat vivamus. Id diam vel quam elementum pulvinar.

Active Release Documentation  
EEL 5.1.0 

Want to Contribute?  
**CONTRIBUTE**

Contributors are needed to keep the AGL project moving forward.